PRIVACY PROTECTION OF EMPLOYER LOCATION USING CASPER QUERY PROCESSING

by

Swagata Mukherjee

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 10, 2016

Keywords: employer location protection, spatial query, location privacy, location crowdsourcing

Copyright 2016 by Swagata Mukherjee

Approved by

Wei-Shinn Ku, Assistant Professor, Department of Computer Science and Software Engineering, Auburn University
Xiao Qin, Assistant Professor, Department of Computer Science and Software Engineering, Auburn University
Sanjeev Baskiyar, Assistant Professor, Department of Computer Science and Software Engineering, Auburn University
George T. Flowers, Dean, Graduate School, Auburn University

ABSTRACT

PRIVACY PROTECTION OF EMPLOYER LOCATION USING CASPER QUERY PROCESSING

Swagata Mukherjee

Master of Science, December 10, 2016

Directed by Dr. Wei-Shinn Ku

Protection of location privacy is often an essential requirement for any new or established business organization. When a new store is being set up, the owners may want to protect their exact location in order to prevent their competitors setting up their stores in a nearby location and attracting their customer base. As a store is being set up in a particular location, the owner may want to recruit new employees to work for them. The potential employees of the store may in turn want to know the location of the store before joining. If the owner does not want the store location to be known to everyone, especially their competitors, he may want to opt for location privacy of his store location. The location privacy algorithm may be such that the tentative store area may be known, but not its exact location. So, the potential employees will be satisfied knowing the broader area in which the store will be located, and the competitors will not be able to pin-point the exact location of the store. Many types of location privacy algorithms have been made use of in the past. Nearly all of them deal with location privacy of individuals making queries, while driving on a busy road network (mobile users). Our algorithms will try to take a different approach and provide location privacy to employers who need them (stationary stores or empty land plots). Till date, to our knowledge, no algorithms exists dealing with anonymizing location of: (a) to be constructed stationary building, which can be a departmental store, chain cafeteria, or even a pharmacy outlet, infamous for setting up rival stores near each other. Here we propose to use the Casper framework approach which can used to provide location privacy of new store employers. This approach takes into consideration that the store has not been constructed yet and all we have to protect is the location of an empty piece of land where it is being set up. We take into account the fact that there are other empty land slots located around and nearby our slot. This method takes into account that a new store is to be built and there are many other empty available lots available for construction. The exact location is not given out in the advertisement to help protect location privacy. We further consider that other established stores of same brand is located in nearby areas. Let us consider a popular store (example, Starbucks) in a city. A new Starbucks store is to be set up in a new location. The new store will apply location privacy algorithm such that the store will be indistinguishable among other Starbucks stores and empty land plots. The prospective employees will not know that they will be working in a new Starbucks store, not an old existing one. This algorithm is applicable for both big cities and smaller towns.

ACKNOWLEDGMENTS

The author would like to express her gratitude to almighty God for guiding her through the entire work spiritually and showing the right path throughout her tenure as a Masters graduate student. This work would not have been possible without Gods help and blessings. The author would also like to express her heartfelt gratitude to her major professor Dr. Wei-Shinn Ku for his support, mentorship, flexibility, encouragement and personal care during the course of this study. The author would concurrently like to express utmost thankfulness towards Dr. Xiao Qin and Dr. Sanjeev Baskiyar for agreeing to serve as committee members on her thesis work. The author would like to extend her gratitude to Dr. Ku and Dr. Marghitu for the teaching assistance provided for most parts of her graduate study at Auburn University. The author would also like to thank Dr. Asif Iqbal Baba for his inputs in the thesis work. The author is also grateful towards all her co-workers and lab members for their help in understanding difficult scenarios when and where they arose. The author would also like to thank her family members who provided moral and spiritual support during the pursuit of this academic goal. In this regard the author would like to thank her parents Mr. and Mrs. Mukherjee, in laws Mr. and Mrs. Bhattacharya, husband Dr. Bhattacharya for their continuous encouragement and support.

Style manual used Auburn University Thesis and Dissertation Template (together with the ”aums” style).

Computer software used LaTex for Windows 64-bit MiKTeX (version:basic-miktex-2.9.6069-x64) together with departmental style file aums.sty.

The experiments were run on a MacBook Pro OS X El Capitan (version 10.11.6), 2.2 GHz Intel Core I7.

The graphs were plotted with Graphpad Prism (version: 5.0)

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

Location based services are used by millions of users on a day-to-day basis. These services have gained popularity in the past decade with the ever increasing popularity of cellphones and GPS (Global Positioning System). Users can query for the location of the nearby Points of Interests POIs and get the results they want within comparatively shorter amount of time. Users query for 'find the nearest restaurants', 'find the nearest gas station', 'find people nearest to me who wants to sublease their apartments', etc. This search process requires the users to disclose their exact locations to their service providers, who then process the query to return the list of POIs the user is querying for. This method has its drawbacks on privacy related issues. The service provider may be vulnerable to hackers and the users personal data, such as his current location can be compromised. This may lead to serious privacy threats such as tracking and stalking.

Much research and resources has been dedicated till date on the privacy protection of mobile users who are using location based services. The most common example of such cases is a user driving down the highway uses his mobile pone to query for a location based POI. The location privacy model used tries to protect the real-time location of this moving user, their location will be cloaked or hidden in such a way that the service provider will be able to return an approximately accurate list of POIs without having to know the exact current location of the user. We will be discussing some of the privacy protection models in the current study.

To protect the location information of the mobile users, Mokbel, et al., 2006, proposed a framework for location based services with location privacy by using the K-anonymity concept discussed in [1]. The K-anonymity concept could be implemented by two main solutions, the centralized spatial cloaking and the decentralized spatial cloaking.

The centralized spatial cloaking mechanism has a trusted server which has all the user information and is responsible for cloaking the users' locations. It will send the cloaked spatial region and the location dependent user queries to the service provider and return the query solutions from the service provider to each individual user. Since there will be many different queries from different users sent to the trusted server at the same time, it can be a performance bottleneck and for each new user query the cloaked region has to be renewed in accordance with the privacy requirements of the user.

The decentralized spatial cloaking is dependent on the collaboration of the user's peers. The peers can share their location information among themselves and create a cloaked region, eliminating the need for a location anonymiser, preventing bottleneck.

This thesis is based on a different user base than a mobile user client who are on a constant move and require dynamic update of the user location. Our user base are the buildings with fixed locations and our scenario is that they have to construct a new building and want to hire employees keeping their own locations private for the time being.

We will use a centralized spatial cloaking mechanism to find the solution to our problem of anonymizing such locations of to be set-up or constructed new organizations which can be a chain of consumer products or a cafeteria or even a corporate line with competitors in the same field. A location anonymizer will cloak the region in which the building would want to be hidden, which will be sent to the privacy-aware query processor embedded in the location based database server. The query processor will determine the most appropriate set of candidates from the cloaked region and send it back to the location anonymizer, who in turn will return the candidate list to the building (employer).

The remaining part of this thesis is organized in chapters. Chapter 2 discusses about the threats of privacy infringement while using location based services. Chapter 2 surveys the related algorithmic solutions to protect the location privacy of mobile users using location based services. Chapter 3 introduces the problem which the thesis work aims to address and the assumptions made to the solution. Chapter 4 discusses the algorithmic design to solve the problem presented in chapter 3. Chapter 5 has the experimental procedure and data used for experimentation. Chapter 6 showcases the simulations and results obtained from the experimentation. Chapter 7 concludes the thesis and discusses the future work that can be done on the open issues. All references used are listed in the bibliography section at the end of the thesis.

Chapter 2

RELATED WORK

There are various types of location privacy models which can be used for hiding the identities of users.[5]

Landmark based privacy model:

The landmark-based location privacy model, which provides location of a POI near to the privacy-protected user. We cannot use this model as it is not very effective for privacy protection. There will be a limited number of points around the landmark, and the location of the query processor will be easily compromised.

Grid Formation:

Another interesting approach for location privacy would be to divide a given area into many smaller grids. It will provide location for a few other grids, but not the user grid. We can also divide location into grids and send info of other users one by one to confuse the hacker.

Pseudonym:

Another method used in the field of location privacy is to replace user by user pseudonym. We can make the user anonymous by giving him a pseudonym (an untraceable ID).

This method needs frequent changing of pseudonyms to prevent malicious parties from accumulating data.

Multiple false locations:

We can also append multiple false locations to a true location report. The location-based service responds to all the reports, and the client picks out only the response based on the true location. False reports are primarily useful when the user submitting them can filter out all but the servers response to a true report.

Obfuscation:

Obfuscating the user location is an useful method of location privacy[2]. It is degrading the quality of location measurements may reduce threats to location privacy. The different types of obfuscation are: Inaccuracy means giving a measurement different from the actual location; Imprecision means giving a plurality of possible locations [3]. Another method is adding noise, for example, adding Gaussian noise with a standard deviation of a certain distance.

Spatial and Temporal Degradation:

Degrading the quality of location measurements may reduce threats to location privacy.

Specialized Queries:

Queries and answers can only be decoded by a privacy-safe query processor.

Due to recent advances in wireless communication technology, mobile devices (e.g., smart phones, tablets, etc.) with Internet access and positioning chips are significantly increasing in popularity.

Anonymity:

The most prevalent used location privacy is the k-anonymity model. The user reports a region containing k-1 other people. K-anonymity is when a person cannot be distinguished from k-1 other people. The k people form the anonymity set. The regions could be any shape so long as they contain k people, meaning an attacker could not know for sure which of the k people reported the location. The time stamps on location measurements could be similarly ambiguous. There are two types of K-anonymity used - the centralized spatial cloaking and the peer-to-peer spatial cloaking [6] [7]. The centralized spatial cloaking is used by the CASPER model, which is a query processor for location databases that lets a user specify a guaranteed k and the minimum area within which he or she wants to hide. The peer to peer spatial cloaking is a cloaked region between trusted peers where no third-party (central server) is present. Each peer has two values at a point in time: k and Amin [8]. The peers communicate between each other. Each peer communicate with service provider directly and anonymously.[9]

Now that we have had a look at most of the location privacy models out there, let us focus on the model we are inspired from to write our thesis. We will discuss in detail about the CASPER's K-anonymity centralized spatial cloaking system in this section.

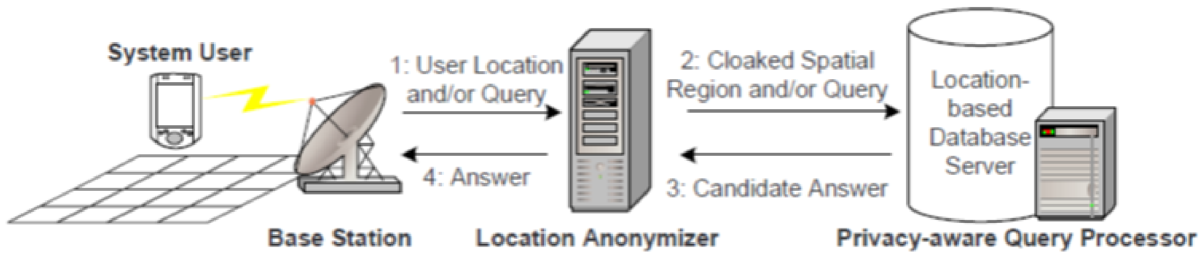

System User
1: User Location and/or Query
4: Answer
2: Cloaked Spatial Region and/or Query
3: Candidate Answer
Location-based Database Server
Base Station
Location Anonymizer
Privacy-aware Query Processor

Figure 2.1: Casper System Architecture

2.1 CASPER System Architecture

Mokbel, Chow and Aref[4] proposed the Casper system architecture [1] as an example of the centralized spatial cloaking mechanism. In this system, every user will have their individual privacy profile defined as a tuple(k, Amin)containing the cloaking requirements of the user: k represents the numbers of other users among which our user wants to be anonymous and Amin is the minimum area of the region to be cloaked by the location anonymizer to hide the user. Larger values of k and Amin represents the user has stricter privacy requirements. Amin is particularly useful in densely-populated regions when larger k value would not be that useful to keep the user location private; k is most useful in sparsely populated areas where larger Amin values would not satisfy user privacy needs.

The Casper system architecture consists of two main components: the Location Anonymizer and the Privacy-Aware Query Processor.

2.1.1 The Location Anonymizer

The location maintains the user's location information and privacy profile. The user's privacy profile consists of a tuple(k, Amin), which the user can change at will. The location anonymizer receives the query request from the user and performs cloaking procedures to anonymize the user location within a bigger region based on the user's privacy profile requirements. It then sends the cloaked spatial region to the query-processor embedded in the location service provider. Once the query-processor processes the query and returns the candidate answer sets, the location anonymizer will return the answer set to the individual user. There are two kinds of location anonymizers used for the centralized spatial cloaking method: the basic location anonymizer and the adaptive location optimizer.

The Basic Location Anonymizer

Data Structure - Figure 2.2 depicts the most common data structure of a basic location anonymizer, which is a grid-based complete pyramid of H levels where a level of height h will have 4^h grid cells. The root ell is the upper-most cell of the pyramid structure represented by cell 0 and covers the entire system area. Each call is represented by (cid, N) where cid is the unique cell identifier and N is the number of mobile users present within that cell. The loation anonymizer also keeps track of a hash table which contains a (uid, profile, cid) for each mobile user. The uid is the user identifier, profile is the user profile and cid is the grid cell in which the user is present, which is always in the lowest level of the pyramid structure for the basic location anonymizer data structure. The location anonymizer

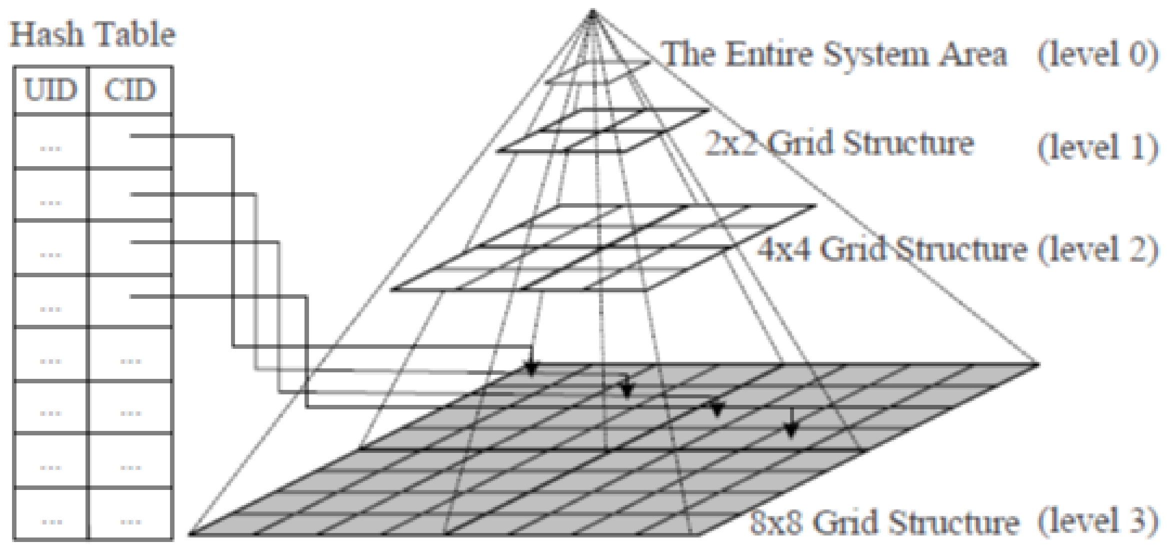

Hash Table
UID
CID
...
The Entire System Area (level 0)
2x2 Grid Structure (level 1)
4x4 Grid Structure (level 2)
8x8 Grid Structure (level 3)

Figure 2.2: Basic Location Anonymizer

dynamically stores the user location in it.

A major drawback of this data structure is the high cost incurred from constant location updates and cloaking time. The mobile user location is dynamically updated to the location anonymizer with respect to the other users' locations. The location anonymizer has to perform a seperate cloaking operation for the individual user if their location is changes or a different user query is issued. These issues are handled in the adaptive location optimizer data structure.

Maintainance - Location updates are received frequently by the location anonymizer whenever there is a change in the users current location, or a user enters or leaves the system. New location of the user is sent in the form of (uid, x, y), where uid is the user identifier and the user's new location spatila coordinates are given by x and y. The location anonymizer receives the new location update and applied hash function h(x,y) to find the new grid cell identifier cid_{new} in which the user is currently located. The cid_{old} is obtained from the previous user information entry in the hash table. No updating is required if cid_{old} is equal to cid_{new}. If cid_{old} and cid_{new} are not same, the user entry in the hash table is updated with cid_{new} and the number of users N in in both cid_{old} and cid_{new} are updated and this change is propagated to all levels in the pyramid.

Cloaking Algorithm - The cloaking algorithm is depicted by Algorithm 1. The algorithm takes the tuple(k, Amin) as input, where k and Amin are the number of users and the area within which the user will be anonymous. The algorithm initially calculates if the the area and k sent by the mobile user can be satisfied within the grid cell(cid) the user is currently present; if so, the area if cell cid is retuned. If not, the algorithm adds this area to the adjascent vertical and horizontal cells in turn, and return s the area which closely satisfies the user privacy profile. If the area is still not found, the algorithm jumps to the parent cell of cell cid and repeats the process until the required area is found.

The Adaptive Location Optimizer

Data Structure - In the adaptive location anonymizer data structure, the cell content and hash table is same as the basic location anonymizer, the main difference is that an incomplete pyramid structure is maintained, as opposed to the complete pyramid structure for basic location anonymizer. The incomplete pyramid in this location anonymizer maintains only those grid cells which will be used for cloaking purposes, and disposes of the rest, so the structure of the pyramid is maintained in real-time based on the current locations and

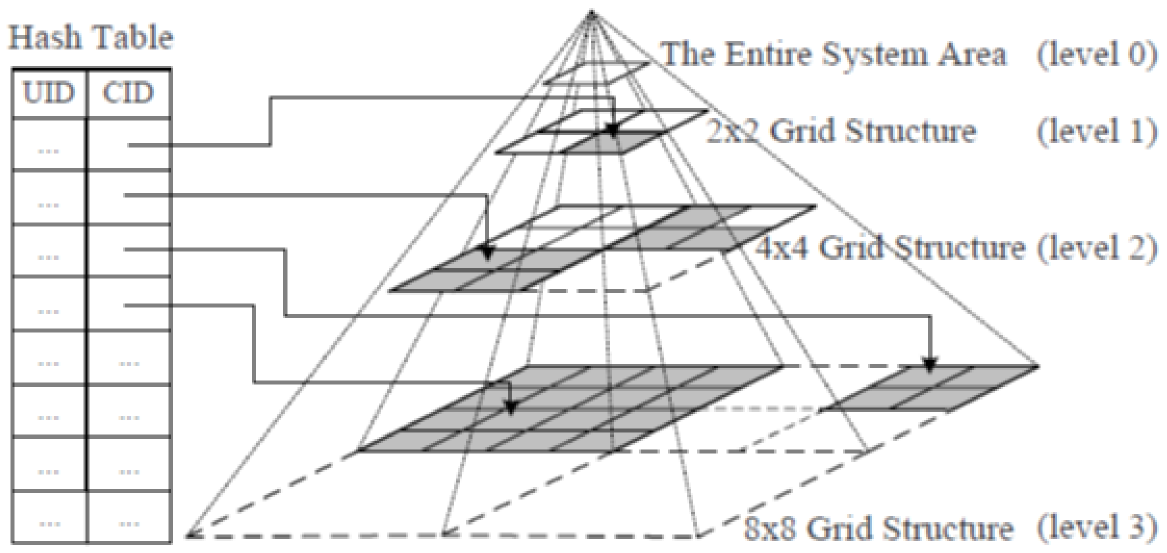

Hash Table
UID
CID
The Entire System Area (level 0)
2x2 Grid Structure (level 1)
4x4 Grid Structure (level 2)
8x8 Grid Structure (level 3)

Figure 2.3: Adaptive Location Anonymizer

privacy requirements of the users.

Maintenance - The adaptive location anonymizer performs all maintenance function of the basic location anonymizer along with the dynamic maintenance of the incomplete pyramid structure. Two main operatons are done to maintain the efficiency of the incomplete pyramid structure: Cell splitting and cell merging.

Cell Splitting: A cell cid at level i are split into 4 cells at a lower i+1 level if there is a user u in cid whose privacy profile can be satisfied by some cell at level i+1.

Cell Merging: 4 cells at level i are merged into 1 cell at higher level i-1 if all users in the cell has strict privacy requirements which cannot be satisfied within level i.

Cloaking Algorithm - The cloaking algorithm of adaptive location anonymizer is different to the basic one in the sense that the algorithm inputs cell cid from the lowest maintained level instead of the lowest pyramid level, which reduces the number of recursive calls made to the algorithm.

2.1.2 Privacy-Aware Query Processor

The location anonymizer sends the cloaked region to the privacy-aware query processor, which is present in the database server. The query processor will receive a cloaked spatial region containing the user region and the user query and return a list of candidate answers. The query processor will never know the exact location of the user during this entire process. Because the query will not contain the exact location coordinates for the user but a cloaked region, the processor will not able to provide the exact answer but a list of candidate points.

The query processor algorithm will calculate a spatial region based on a few filter and return the candidate list included in that region. The list returned will be fairly accurate even though the processor was unaware of the exact user location.

There are 4 steps a query-processor follows to get the candidate answer list. The first step is the filter step, where four filter target objects are chosen as the nearest point t_i for each vertex v_i in the cloaked region A.

The second step is the middle point step, where we try to find the point m_{ij} for each edge $e_{ij} = v_i v_j$ such that m_{ij} divides e_{ij} into two segments $v_i m_{ij}$ and $m_{ij} v_j$. The basic idea is that any point in the first segment $v_i m_{ij}$ will have t_i as its nearest filter target object and any point in the second segment $m_{ij} v_j$ will have t_j as its nearest filter target object while point m_{ij} is of equal distance from both targets t_i and t_j . We distinguish between two cases based on whether the two vertices v_i and v_j have the same filter or not. If v_i and v_j have the same filter t, then point m_{ij} does not exist as all points on edge e_{ij} will have t as their nearest target object. If t_i and t_j are different, m_{ij} is found by connecting t_i and t_j through a line L_{ij} . Then, another line P_{ij} is plotted that is perpendicular to L_{ij} and divides L_{ij} into two equal segments. Finally, m_{ij} will be the intersection point between P_{ij} and the edge e_{ij}.

The third step is the extended area step, in which for each edge e_{ij}, we will try to find out the largest distance max_d from any point on E_{ij} to its nearest filter target object. Only three points on e_{ij} can be candidates to have the distance max_d to their nearest filter object, v_i, v_j, or m_{ij}. Thus, we compute the three distances d_i, d_j, and d_m that represent the distances from points v_i, v_j, and m_{ij} to targets t_i, t_j, and t_i. If m_{ij} does not exist, the distance d_m will be 0. Then, the distance max_d is computed as the maximum distance of d_i , d_j and d_m. Finally, the area A_{EXT} is expanded by the distance max_d in the same direction of edge e_{ij}.

The last step of this algorithm is the candidate list step, where the server issues a range query that returns all target objects within the area A_{EXT} as the candidate list. The candidate list is sent to the client where the query can be evaluated.

Chapter 3

PROBLEM STATEMENT AND ASSUMPTIONS

The problem that this thesis is aiming to solve is how to protect the location information of a to-be-built store such that the employer can find eligible future employees nearby to work for them without risking exposing his store location to the competitors before the store is up and running.

Retail market for department stores, pharmacies, chain restaurants is full of competition. New stores are coming up rapidly and every corporate giant wants to grab the market from its competitors and make the most profit. They try to set up new store at every corner of the map, but whenever they set up the new store, their competitors set up their store near to the location to attract their customer base. This can be very frustrating to these companies.

This can be seen when we take the real-life competitor stores into consideration. Wherever there is Walgreen's, we'll find a Rite Aid or CVS store nearby. We'll fins a target very near to a Walmart store. If we take chain restaurants into consideration, we'll find a MacDonald near to a Burger King. Such examples are abundant in the modern world when there are so many companies around fighting over the same customer base to make more profit than their competitors.

Our solution is to hide the store location when the store is being constructed, to be able to fend competition a little longer. Once the store is opened to public, the location ca get disclosed and the competitors may start to swarm the place, but by the time competitors get a foothold nearby, our store will be flourishing, already established and making profit.

This problem can be solved by a modified version of the Casper system architecture which is originally used for mobile users. Our version will take into account the fact that the building location is fixed, so the pyramid structure maintenance of the location anonymizer does not have to be dynamic. Our store has to be anonymous among k other POIs consisting of other stores and empty land plots.

When the query is being sent for candidate locations, it is not mentioned that the requirement is for a to-be constructed store or a pre-existing one. The empty land plots need to be included in our system such that if a hacker hacks into the system architecture, they will see a list of stores and empty land plots and will not be able to figure out that the employment request is being sent on behalf of a future store. On the contrary, if we try to hide our store location among k other existing stores and a hacker hacks our system, they will get a list of k stores and exactly 1 empty land plot, which will make it very easy for them to know a new store is being constructed at that particular empty land plot.

Another advantage of adding the empty land lots to the k POI is that it increases the density of k in any given area, hence requiring lesser cloaking area, speially in regions which are sparsely populated.

Chapter 4

DESIGN

4.1 Modified Casper System Architecture

The store will register with Casper with a privacy profile defined by a tuple(k, Amin), where k is the minimum number of stores or empty land plots within which the store wants to be hidden and Amin will be the minimum acceptable area of the cloaked region covering the store [10]. The store has the option to change their privacy profile as and whenever needed [1].

There are two main components of this system architecture: the Location Anonymizer and the Privacy-Aware Query Processor. Using the basic location anonymizer for our solution is more feasible as our users have fixed locations dynamic updating of incomplete pyramid structure is not required. The location anonymizer blurs the store location under a cloaked spatial region in accordance with the store privacy profile(k, Amin) and sends it to the

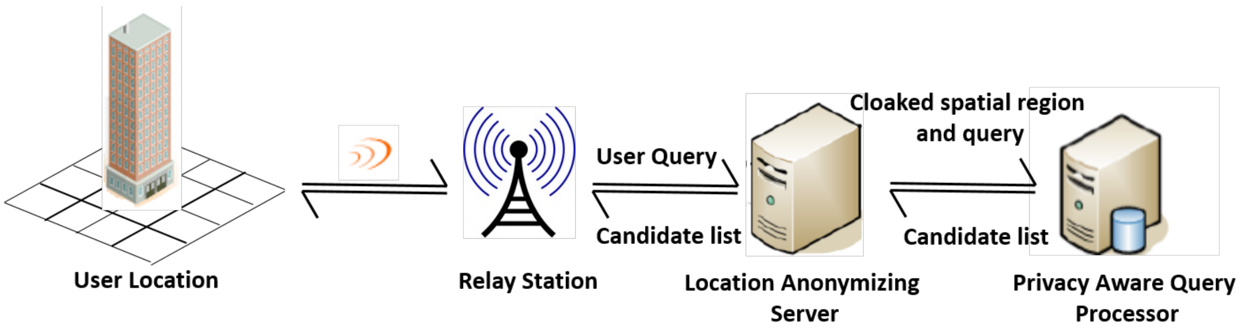

User Location
Relay Station
User Query
Candidate list
Location Anonymizing Server
Cloaked spatial region and query
Candidate list
Privacy Aware Query Processor

Figure 4.1: Casper System Architecture

query processor, along with the query sent by the store. The privacy-aware query processor is embedded in the location-based database server. The query processor receives the cloaked spatial region of the store and its query from the location anonymizer and returns a list of candidates which are most appropriate for the query sent. If the store is querying to hire new employees nearby, the query processor will return the list of candidates who are interested to be hired in the region.The candidate list is received by the location anonymizer and sent to the store for evaluation. Larger k and Amin of the store's privacy profile will return a longer list of candidates for the job. The store manager can then contact the interested candidates without giving away their own location.[11] [12]

4.2 Location Anonymizer

The main function of location anonymizer is to blur the store location within a cloaked spatial region based on store's privacy profile and send the cloaked area and the store query to the privacy-aware query processor. There are two types of location anonymizer used for privacy protection of mobile users: the basic location anonymizer and the adaptive location optimizer. Our solution uses the basic location anomizer. There is no need for us to use the adaptive optimizer as our user base has a fixed location and there is always a single user in each system, which eliminates the need of dynamic maintenance of the incomplete pyramid structure required for the adaptive one.

The basic location anonymizer satisfies the 4 main features reqired for it to satisfy our

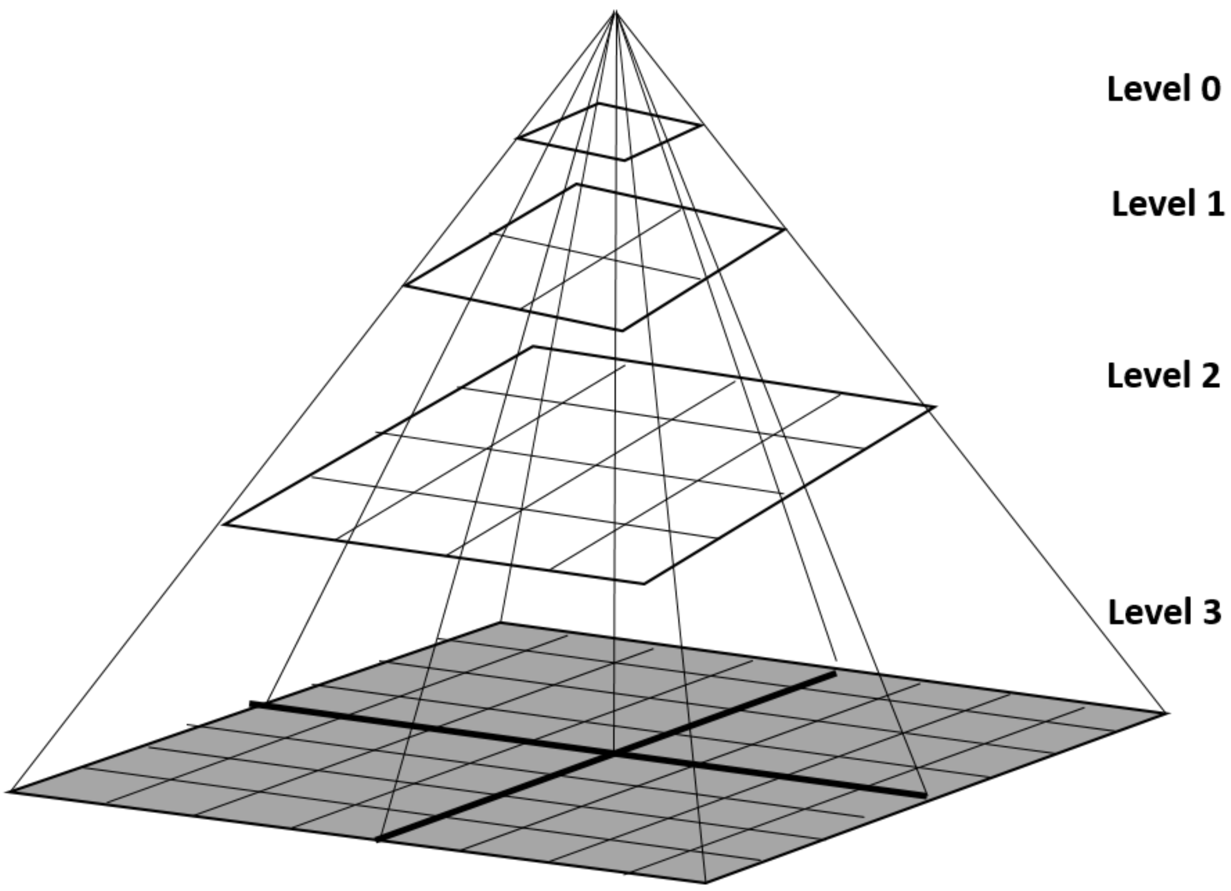


Figure 4.2: Basic Locaton Anonymizer Data Structure

solution: Accuracy, Quality and Efficiency. The cloaked spatial region obtained satisfies the store's privacy profile(k, Amin) [13]. It is not possible for someone to know the exact store location within the cloaked region containing both stores and empty land lots. The cloaking algorithm consumes less CPU time and memory compared to the dynamic location anonymizer, as the cloaking effort is only based on a single store's need, which is fixed in location. Flexibility is not required for the location anonymizer used in our solution. We will now move on to the details of our location anonymizer.

Data Structure

The location anonymizer data structure is a grid-based pyramid that hierarchically decomposes the Euclidean space into H levels where there are 4^h grid cells in a level of height h. So in the pyramid architecture shown in figure 4.2, four lower level cells combine to form one upper level cell. The highest level is the root of the pyramid which covers the entire spatial data region. The root is represented as level 0 and the lower cells are level 1, level 2, level 3, etc. Eah grid cell in the pyramid is represented by a tuple(cid, N), where cid is the unique cell identifier and N is the number of other stores or land lots located within the boundary of that particular cell. The pyramid structure is constant and no dynamic maintenance is reuired, except for when the store may choose to update its privacy profile. A hash table is created when the store is first cloaked, containing (uid, profile, cid) of each store and lot within the entire data region loaded in the location anonymizer. Uid is a unique identifier assigned to each store or empty land plot, profile is only present for the store whose location is be anonymized and cid are the cell identifier given to each cell. Cid will always be at the lowest level in the pyramid data structure, as represented by the shaded region in figure 4.2.

Maintenance

The basic location anonymizer used in this case does not need to be dynamically maintained. When the store sets up its privacy profile, it sends it to the location anonymizer along with its location. the loaction anonymizer keeps a record of the store's location and privacy profile, along with the locations of all the other stores and empty land plots around it.

When a query is sent to the location anonyizer, it will cloak the region around the store

Algorithm 1 Bottom-up cloaking algorithm

```
1: function BOTTOMUP-CLOAKING($k, A_{min}, cid$)
2:     if $cid.N \geq k$ and $cid.Area \geq A_{min}$ then
3:         return Area($cid$);
4:     $cid_V \leftarrow$ The vertical neighbor cell of $cid$.
5:     $cid_H \leftarrow$ The horizontal neighbor cell of $cid$.
6:     $N_V = cid.N + cid_V.N$
7:     $N_H = cid.N + cid_H.N$
8:     if ($N_V \geq k$ OR $N_H \geq k$) AND $2cid.Area \geq A_{min}$ then
9:         if ($N_H \geq k$ AND $N_V \geq k$ AND $N_H \leq N_V$) OR $N_V < k$ then
10:            return $Area(cid) \cup Area(cid_H)$;
11:        else
12:            return $Area(cid) \cup Area(cid_V)$;
13:    else
14:        BOTTOMUP-CLOAKING(($k, A_{min}$), PARENT($cid$));
```

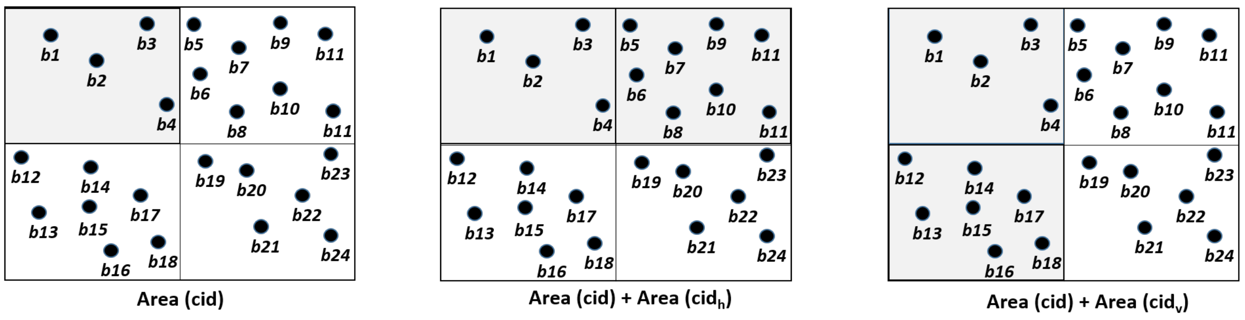


Figure 4.3: Cloaking procedure of Location Anonymizer

based on the previously recorded privacy profile of the store and send the query accordingly. It does not need to keep track of any location change for each store.

Cloaking algorithm The cloaking algorithm of the grid-based pyramid data structure is given by algorithm 1. It is a bottom-up cloaking algorithm.

The algorithm takes the number of stores/lots among which the user wants to be anonymous (k), the minimum area of the cloaked spatial region(Amin) and the cell identifier of the grid cell in which the store is located(cid) as input. k should be less than the total number of stores/lots registered to the system and Amin should be less than the total spatial area, else the algorithm fails.

The algorithm at first checks if the number of stores/lots present in the cell cid is greater than the required number of stores among which our store wants to be indistinguishable, and the area of the cell cid is greater than the minimum spatial area of the required cloaked region, then the algorithm returns the area of the cell cid as the cloaked spatial area.

If the previous condition fails, then the algorithm calculates the area of cid's vertical neighbor (cid_V) and horizontal neighbor (cid_H). To be a neighbor, the cells must have the same parent as cid and lie in the same row or column. If the area of cid with either area of cid_V or cid_H satisfies the privacy profile requirements of the store, the sum of the area which is closest to the user's privacy profile is returned. If not, the bottom-up algorithm is executed recursively with the parent cell of cid, and this goes on until a valid cloaked spatial region is returned.

4.3 Privacy-Aware Query Processor

The privacy-aware query processor is embedded in the location-based database server. Its main function is to return a list of reasonably efficient and accurate list of candidates based on the cloaked area and query provided by the location anonymizer. The database server can store both public and private data. Public data contains persons and facilities who do not want to hide their location information, so they are stored in the database server directly without the need of a location anonymizer. The private data contains personal information which are stored in the database server as cloaked spatial region from the location anonymizer.

The query issued by our store is "find nearby candidate locations interested in store job". The location anonymizer knows the store location when it cloaks the region based on the store's privacy profile, but the query processor is unaware of the exact location information of the store and only receives the cloaked region the store is located in. In our case we assume that the database server is aware of the exact candidate locations who are registered to its system for jobs.

Figure 4.4 shows the candidate location data information stored on the server side. There are 31 candidate locations, c1 to c31 stored in the database server in our example. The candidates who fall directly under the cloaked spatial region sent from the location anonymizer are represented as black circles, and the remaining candidates are shown as grey circles. The location of the store in the cloaked spatial region is shown as a black square, but this locatio is not available to the database server.

Our query processor algorithm initially selects 4 candidate locations as filter objects, which can be used to refine the search over the entire the candidate locations. Theses filter objects can be used to generate a larger area A_{EXT} which contains all the candidate list nearby to the store location. All candidate locations present under ate area A_{ext} will be

Algorithm 2 Private NN Queries over Public Data

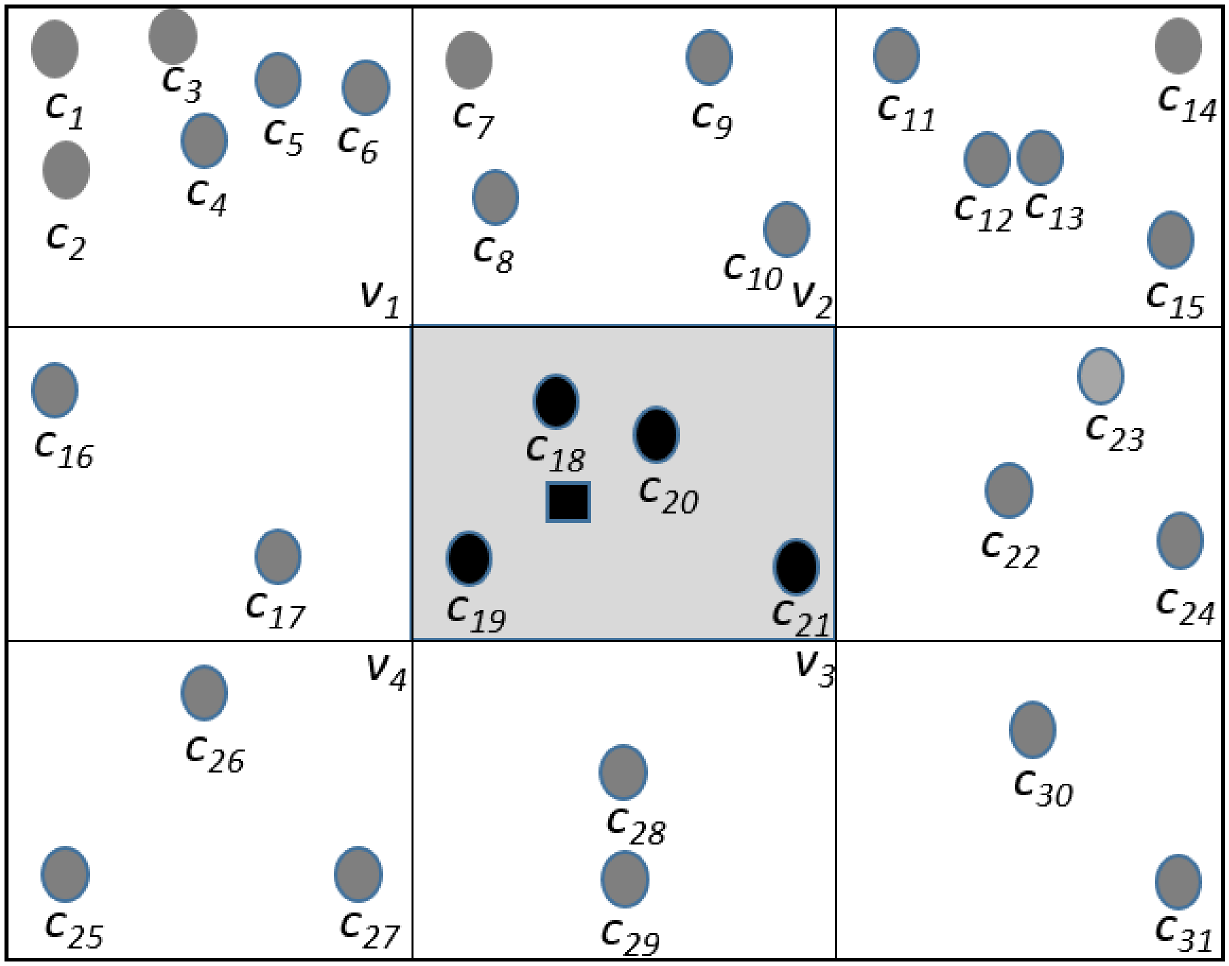
```
1: function PRIVATE-NN-PUBLIC-DATA(Cloaked Area A)
2:     $A_{EXT}$ is an extended area initially set to A
3:     for each vertex $v_i$ in region A do
4:         $t_i \leftarrow$ is the nearest target object to $v_i$
5:     for Each edge $e_{ij} = v_i.v_j$ of region A do
6:         if $t_i = t_j$ then
7:             $m_{ij} \leftarrow$ NULL
8:         else
9:             $L_{ij}$ is a line connecting $t_i$ and $t_j$
10:            $P_{ij}$ is a line that divides and is orthogonal to $L_{ij}$
11:            $m_{ij}$ is the intersection point of $P_{ij}$ and $e_{ij}$
12:        $d_m \leftarrow$ Distance$(t_i, m_{ij})$ = Distance$(t_j, m_{ij})$
13:        $d_i \leftarrow$ Distance$(v_i, t_i)$
14:        $d_j \leftarrow$ Distance$(v_j, t_j)$
15:        $max_d \leftarrow$ MAX$(d_m, d_i, d_j)$
16:        Expand $A_{EXT}$ by by distance $max_d$ in $v_i.v_j$ direction
17:    candidate-list $\leftarrow$ All target objects inside $A_{EXT}$.
18:    return candidate-list
```

returned by the query processor as the required candidate locations.

Algorithm 2 gives us the pseudocode for a privacy-aware query processor. The steps of this algorithm is shown by the figures. Algorithm 2 takes the spatial cloaked area returned by the location anonymizer of algorithm 1 as input. A_{EXT} is initially set equal to the cloaked region A. This algorithm has 4 basic steps as shown by the corresponding figures.

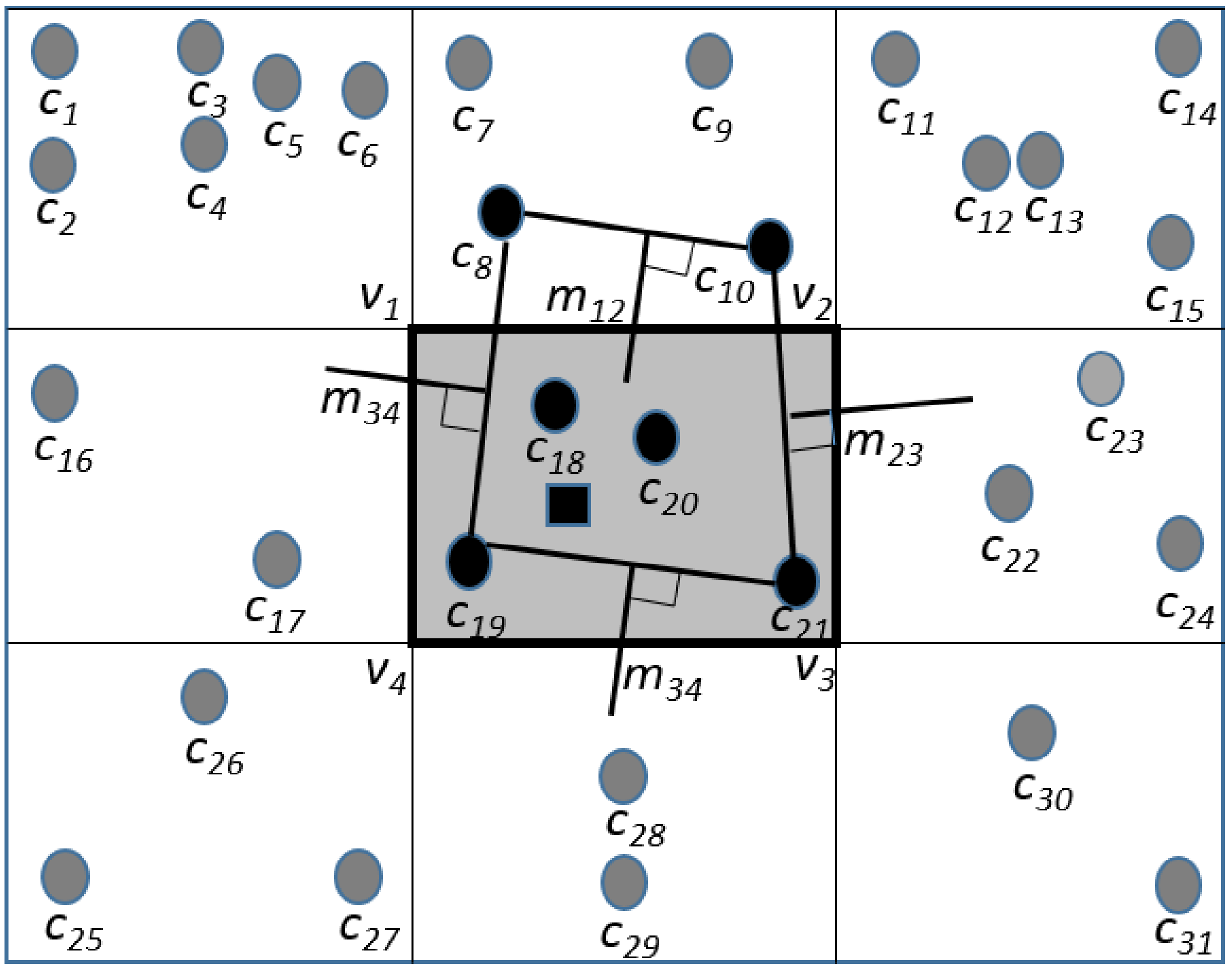
The Filter Step:

We take into account the 4 vertex of the cloaked area and find the nearest candidate location to each vertex, to be used as the filter objects. In our figure 4.4, the candidate locations nearest to vertex $v1$, $v2$, $v3$ and $v4$ are $c8$, $c10$, $c21$ and $c19$ respectively, as we will take them as our filter objects.



a. Step 1. Filters

Figure 4.4: The Filter Step



b. Step 2. Middle points

Figure 4.5: The Middle Point Step

The Middle Point Step:

In this step, we find 4 points on the edges of the cloaked area m_{ij} that divides each edge e_{ij} $= v_i v_j$ of the cloaked area. if both v_i and v_j have the same candidate location as their filter, then m_{ij} cannot be found.

To find the m_{ij} points, we connect $c8$ to $c10$ and from the mid-point of this connected line, we draw a perpendicular line which intersects the line $v1v2$ at point $m12$. $m12$ divides line $e12 = v1v2$ in 2 segments $v1m12$ and $m12v2$ and is of equal distance to candidate locations $c1$ and $c2$. We apply this method to the other 3 edges $v2v3$, $v3v4$ and $v4v1$ to plot the corresponding points on the edges as $m23$, $m34$ and $m41$ respectively, as depicted in figure 4.5.

The Extended Area Step:

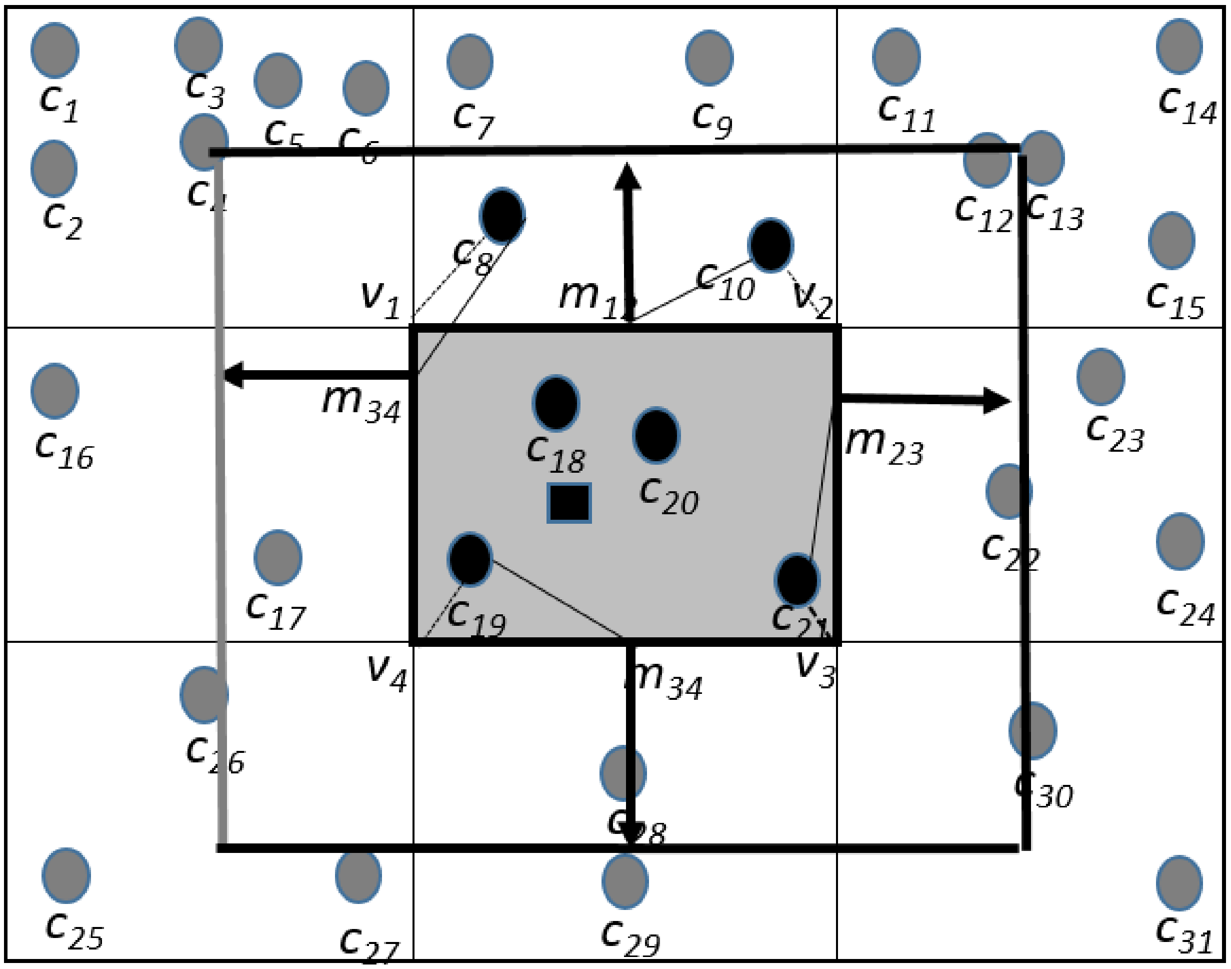
In this step we find the length by which the area would be extended in a particular direction to get the proper A_{EXT} area. The following are calculated first by the algorithm, as shown in figure 4.6

d_m - The distance between c_i and m_{ij} is calculated, which will be equal to the distance between c_j and m_{ij}. If m_{ij} does not exist from the previous step, then d_m will be 0.

d_i - The distance between c_i and v_i is calculated.

d_j - The distance between c_j and v_j is calculated.

We compute max_d as the maximum distance among d_m, d_i and d_j and expand the area A_{EXT} by max_d distance in the direction of edge e_{ij}.



c. Step 3. A_{EXT}

Figure 4.6: The Extended Area Step

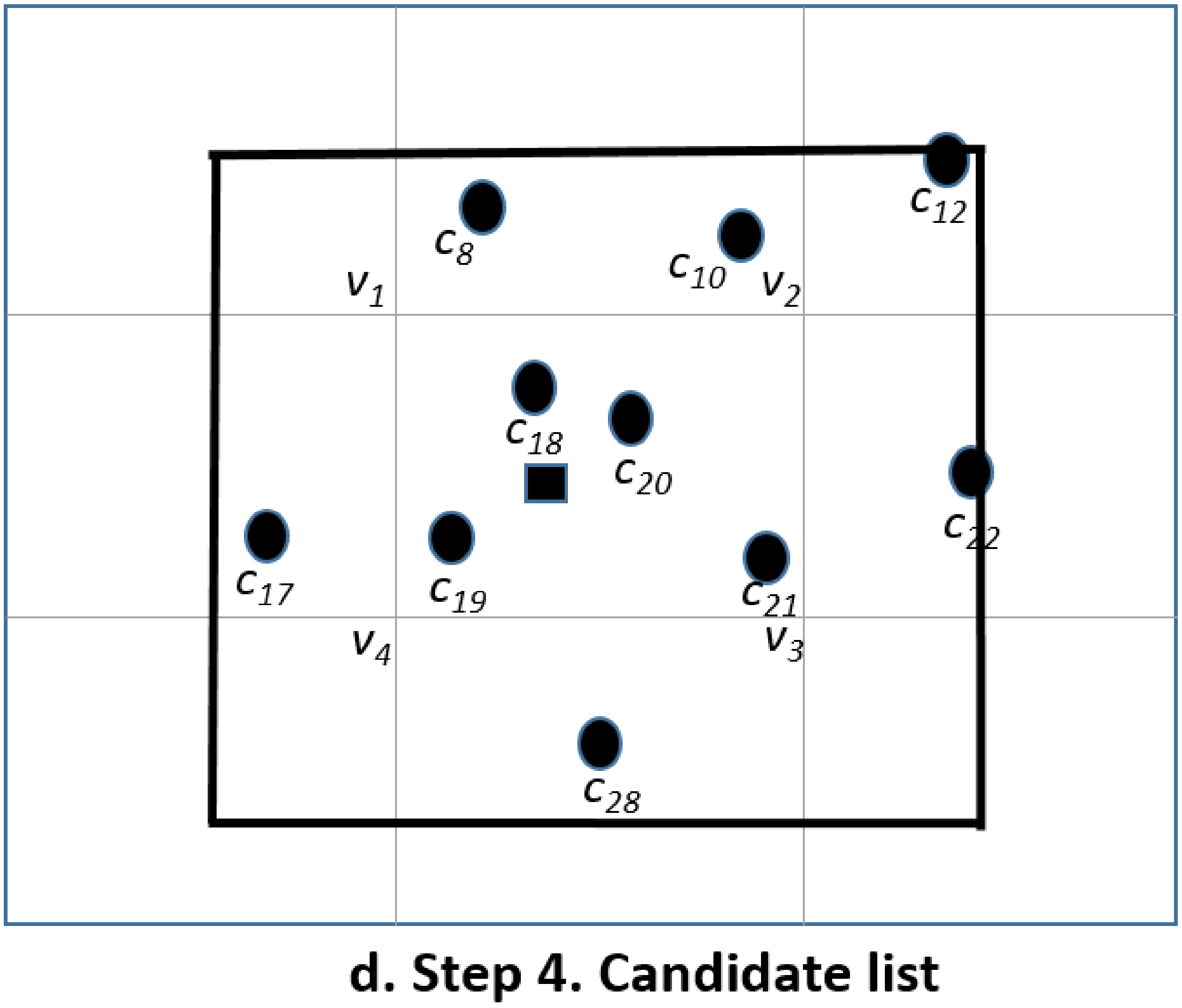


Figure 4.7: The Candidate List Step

In figure 4.6, we see the max_d represented in bold while the other distances are shown dotted. The arrow representing the distance expanded is also shown in bold, in the direction it is to be expanded. A_{EXT} is plot when A_{EXT} is expanded in all directions by the corresponding max_d.

The Candidate List Step:

The query processor now returns all candidate locations present within the A_{EXT} area as candidate list to the location anonymizer. This is shown by figure 4.7.

Chapter 5

EXPERIMENTS

For performing the experimental analysis on our proposed system, a list of locations of different stores and empty land lots available for sale were collected for a few major cities, which include Atlanta, California, etc. The store locations which were used for our experiments are Starbucks, McDonald's, Burger King, CVS Pharmacy, Walgreens Pharmacy, Kroger, Lowe's Home Improvement, Publix, Ross Dress for Less, Target and Walmart. A map of Atlanta containing all Starbucks coffee shop locations (obtained from starbucks.com website) and all empty land lots up for sale(from zillow.com) is shown as a reference 5.1.

The addresses of all stores were geo-coded and converted to longitudes and latitudes for experiment simplification. The longitude-longitude pair were input to a python program which took a modulus of their values and converted them from degrees to radians using the following formula:

$$LAT = (latitude * pi)/180$$

$$LON = (longitude * pi)/180$$

The radian coordinates were then converted to x and y coordinates using the formula:

$$x = R * Sin(LAT) * Sin(LON)$$

$$y = R * Cos(LAT)$$

The distance between the POIs were calculated using the formula:

$$dlon = lon2 - lon1$$

$$dlat = lat2 - lat1$$

$$a = (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2$$

$$c = 2 * atan2(sqrt(a), sqrt(1 - a))$$

$d = R * c$ (where R is the radius of the Earth)

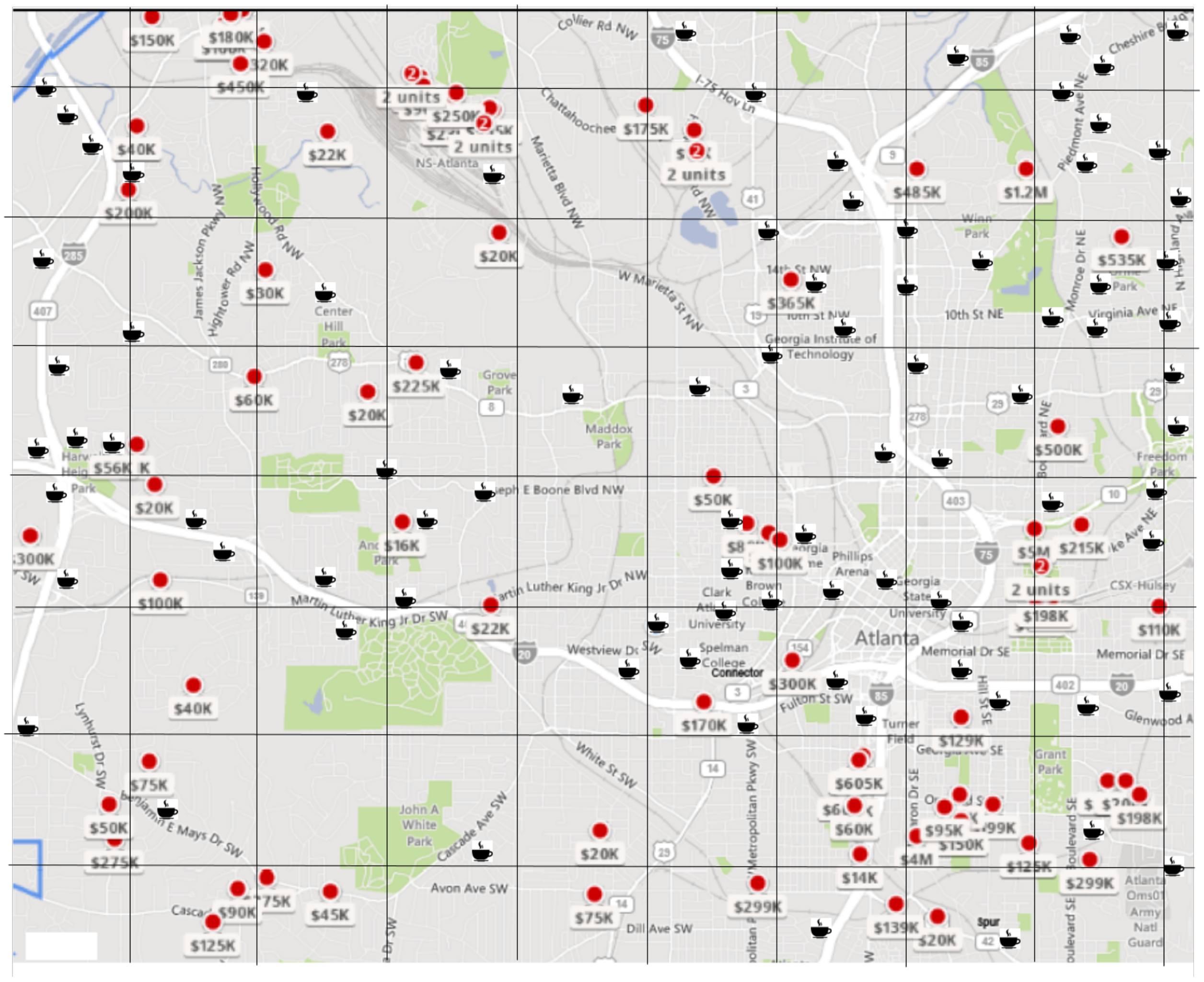


Figure 5.1: Map of Atlanta showing all Starbucks and empty land lots

Chapter 6

SIMULATIONS AND RESULTS

This section experimentally evaluates the performance of Casper system architecture to our specific scenario. For the experiments, we obtained the store location information from the store websites and the land lots up for sale locations from zillow.com. The data obtained were geocoded and converted to the respective x and y coordinates, as already mentioned in the experiments section of the thesis. The candidate locations were depicted to be uniformly distributed in the spatial space.

The graphs were plotted with a professional version of the Graphpad Prism software on a Windows machine. Each experiment was run 10 times and the average of each result was calculated to obtain an accurate graphical plot.

Figure 6.1 shows the increase in the CPU time taken during performing cloaking operations with the increase in the number of stores and land lots within which the store wants to be anonymous.

Figure 6.2 shows the increase in the total CPU time taken by the algorithm from inputting data to returning the list of candidates with the increase in the number of stores and land lots within which the store wants to be anonymous.

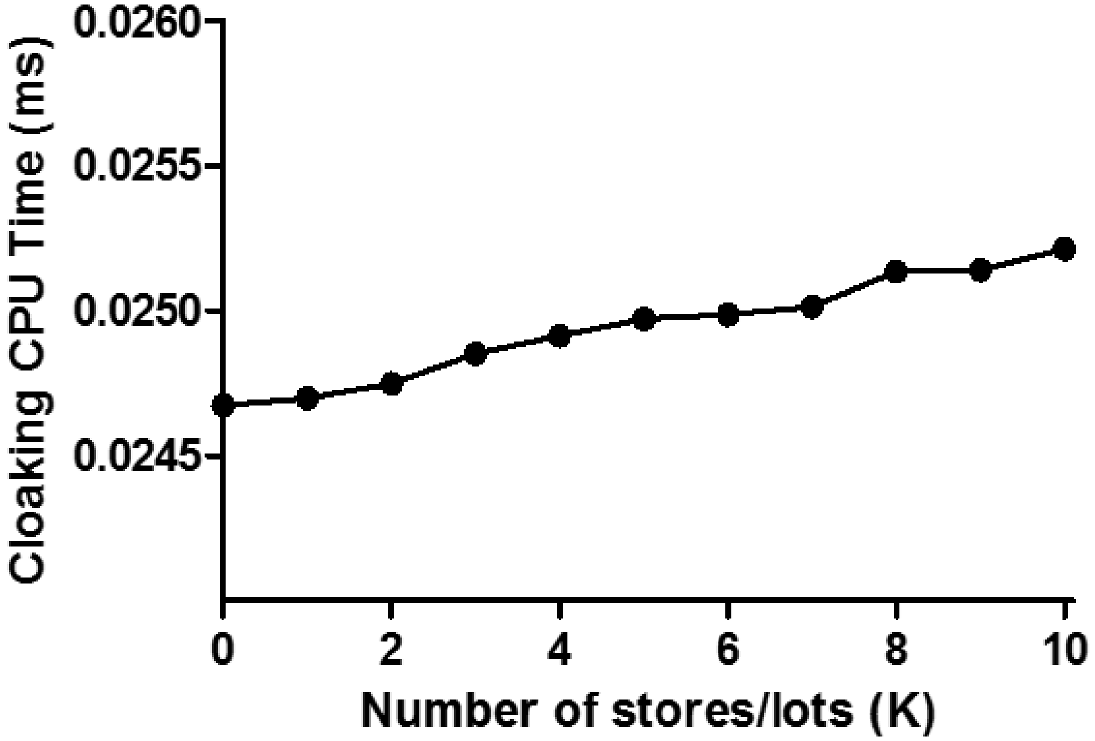


Figure 6.1: K vs Cloaking Time

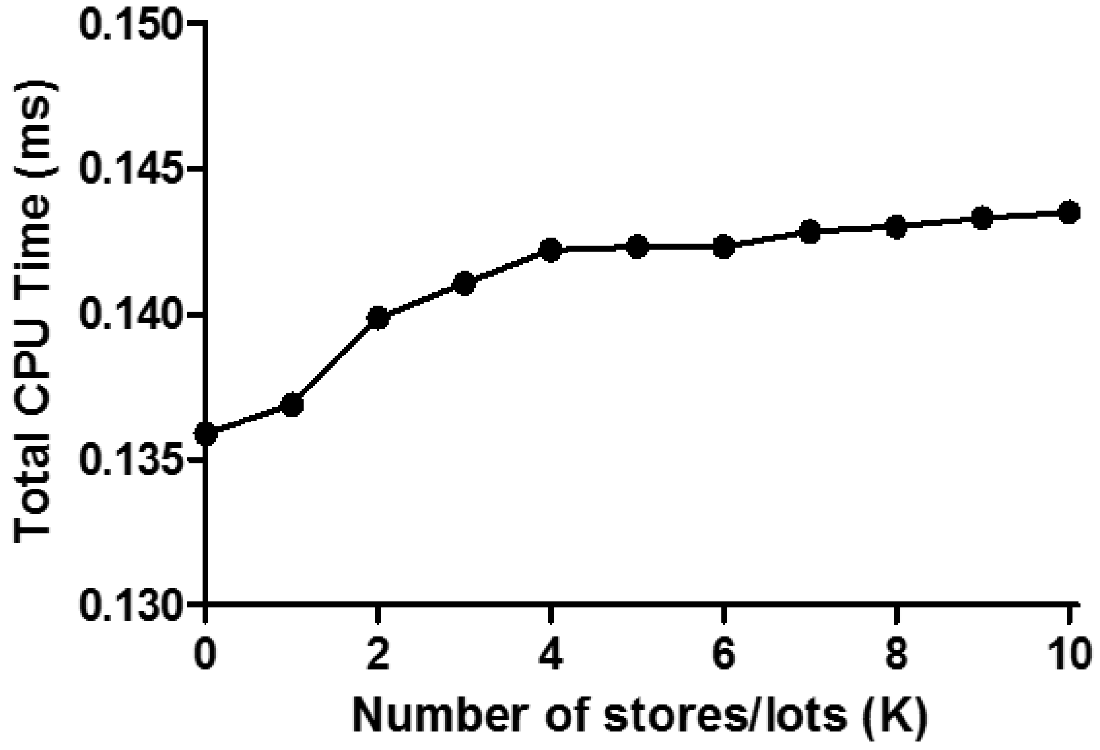


Figure 6.2: K vs Processing Time

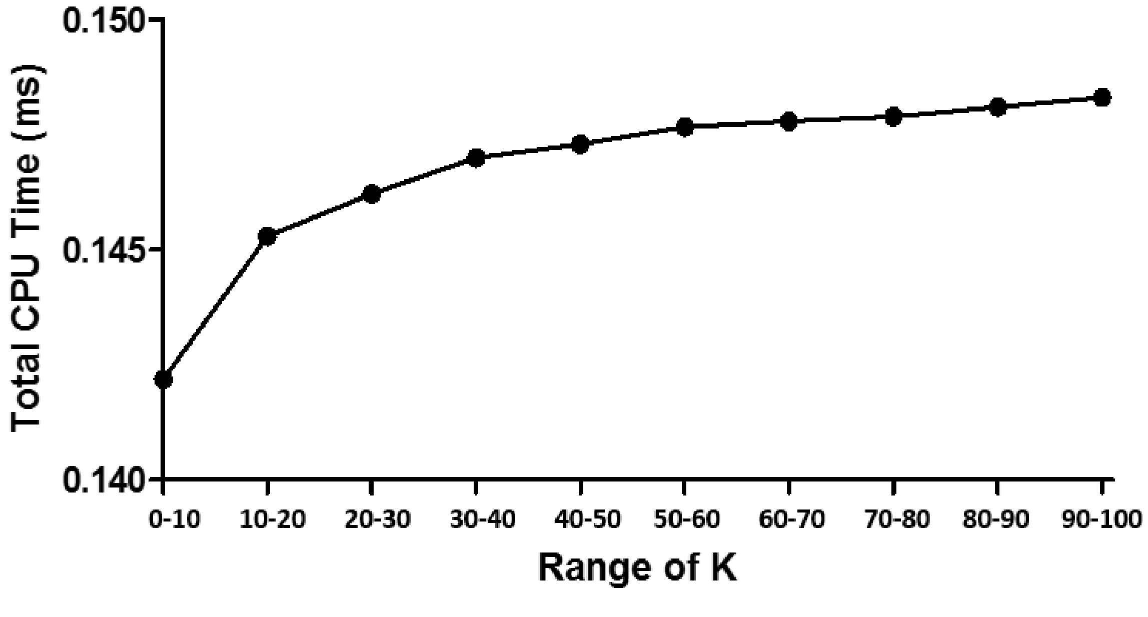


Figure 6.3: K Ranges vs Processing Time

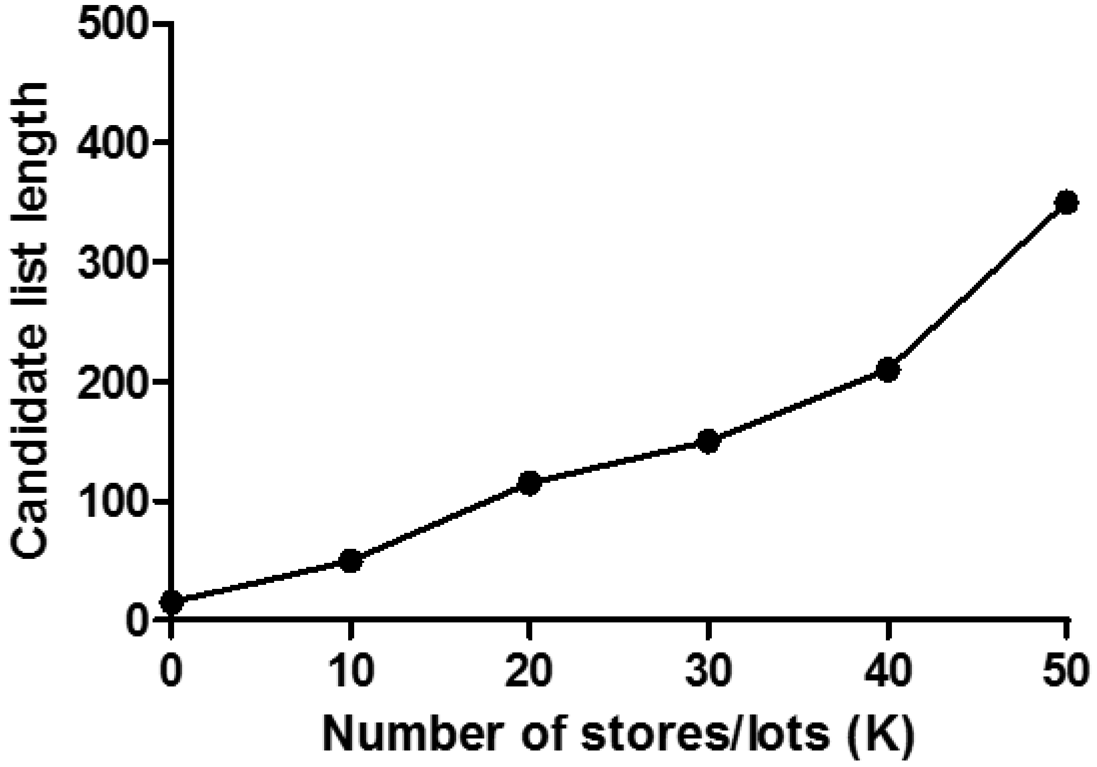


Figure 6.4: K vs Candidate List

Figure 6.3 shows the increase in the total CPU time taken by the algorithm from inputting data to returning the list of candidates with the increase in the range of stores and land lots within which the store wants to be anonymous. Average of each point in the range is taken to find the exact point for that range.

Figure 6.4 shows the increase in the length of the candidate list returned by the query processor with the increase in the number of stores and land lots within which the store wants to be anonymous.

Figure 6.5 shows the increase in the length of the candidate list returned by the query processor with the increase in the spatial area within which the store wants to be anonymous.

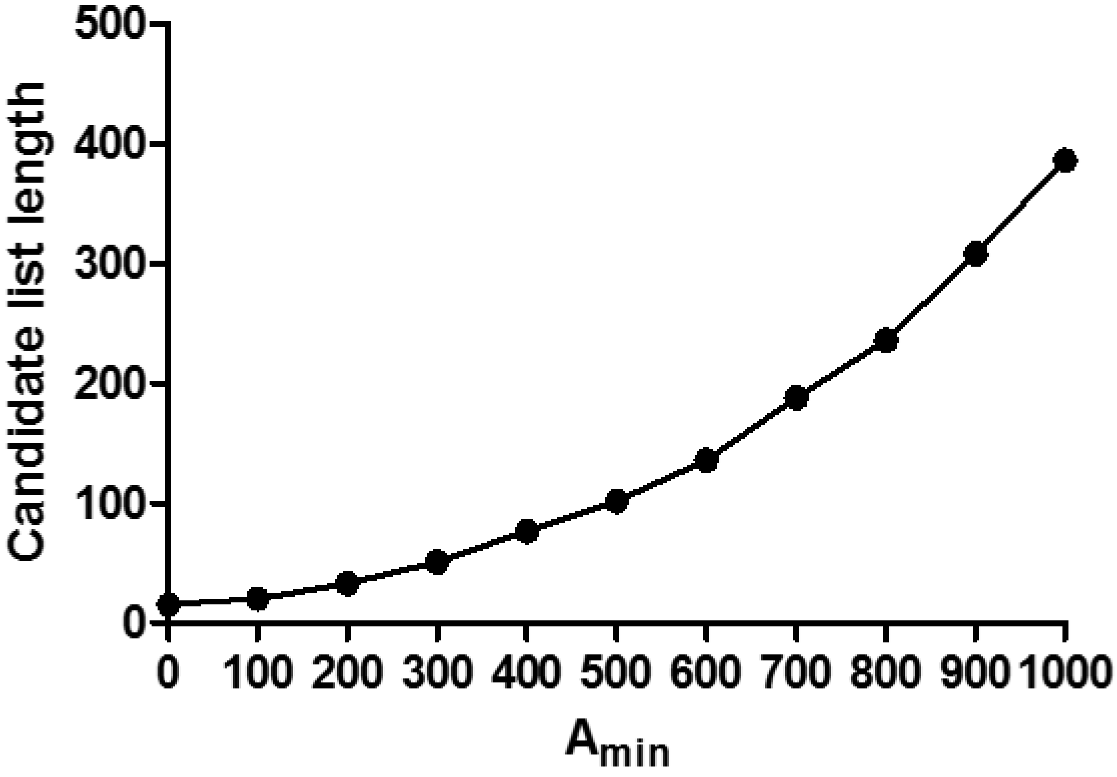


Figure 6.5: Amin vs Candidate List

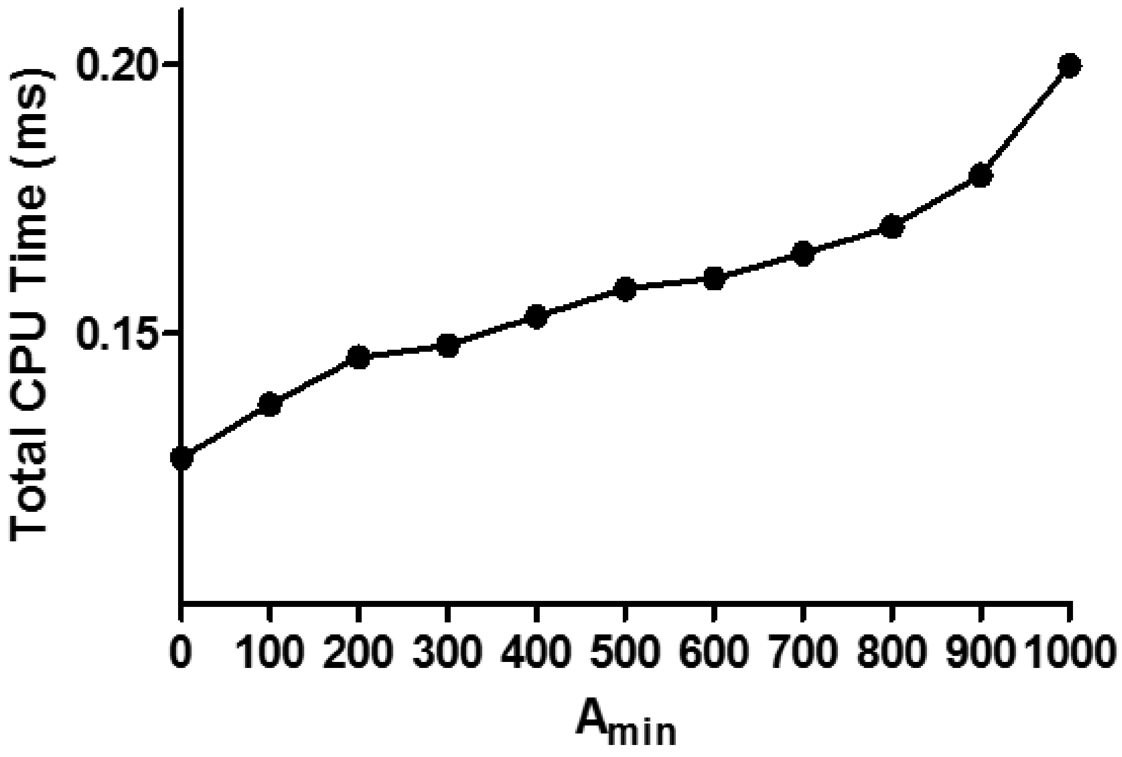


Figure 6.6: Amin vs Processing Time

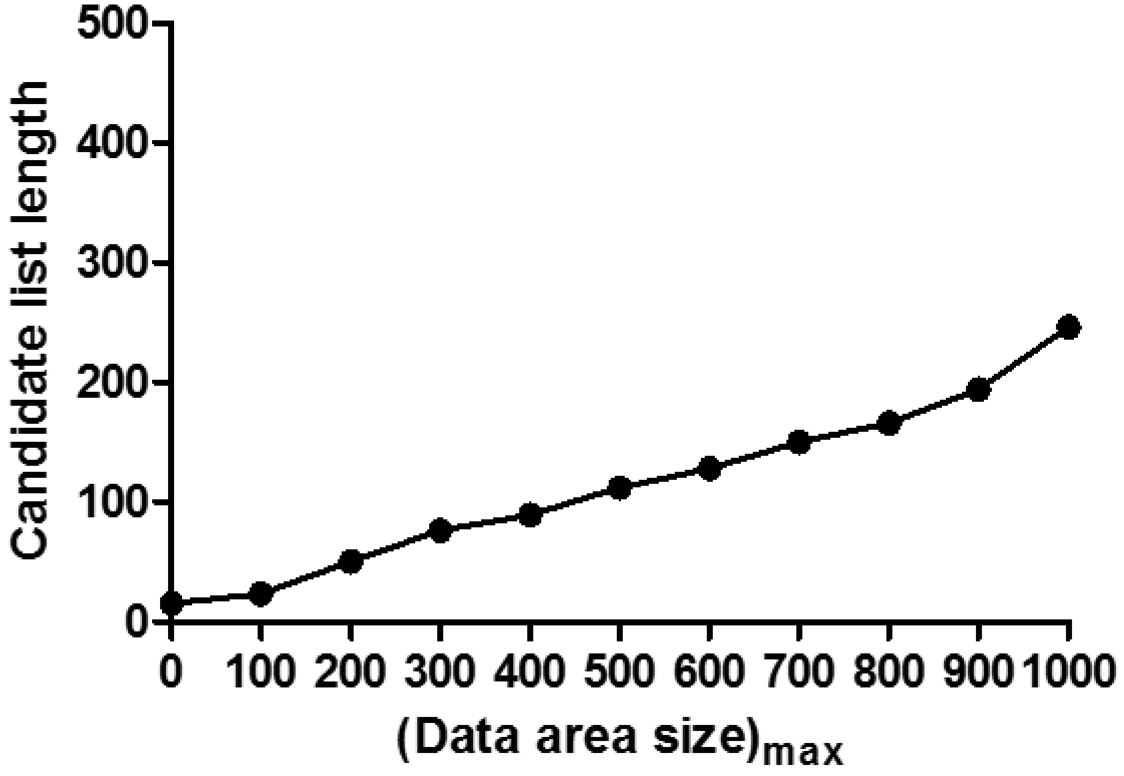


Figure 6.7: Max data area vs Candidate List

Figure 6.6 shows the increase in the total CPU time taken by the algorithm from inputting data to returning the list of candidates with the increase in the spatial area within which the store wants to be anonymous.

Figure 6.7 shows the increase in the length of the candidate list returned by the query processor with the increase in the maximum data area size.

Figure 6.8 shows the increase in the total CPU time taken by the algorithm from inputting data to returning the list of candidates with the in crease in the maximum data area size.

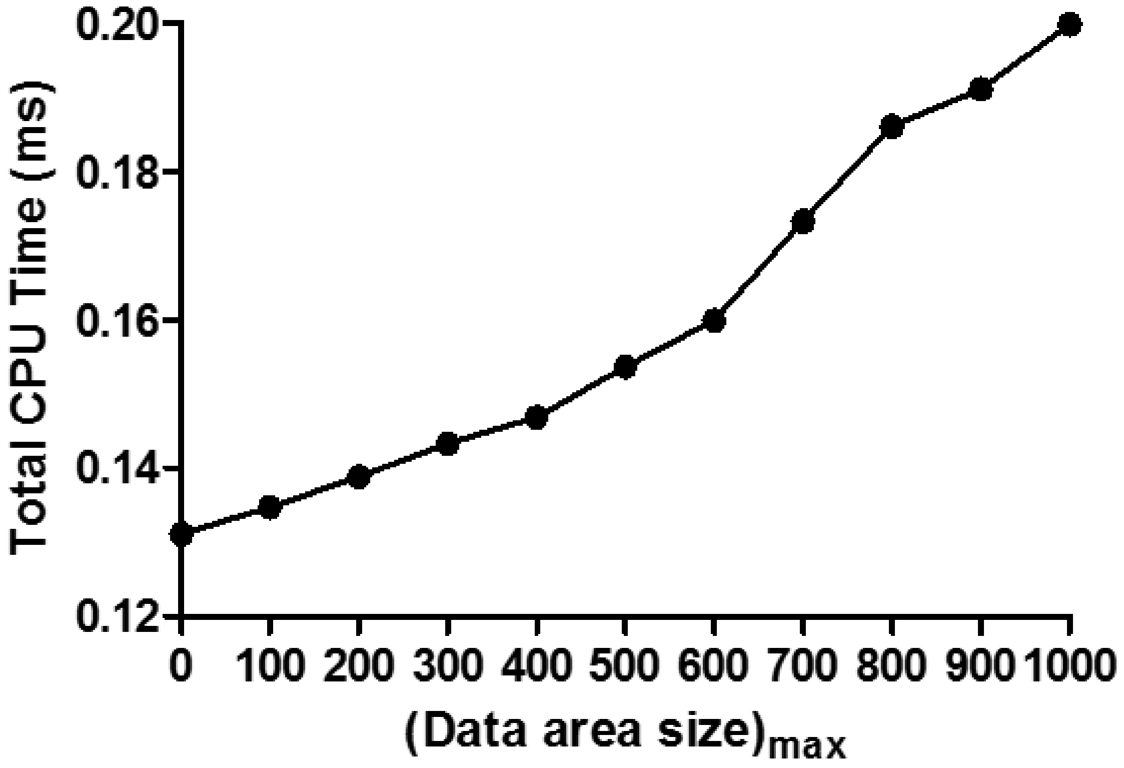


Figure 6.8: Max data area vs Processing Time

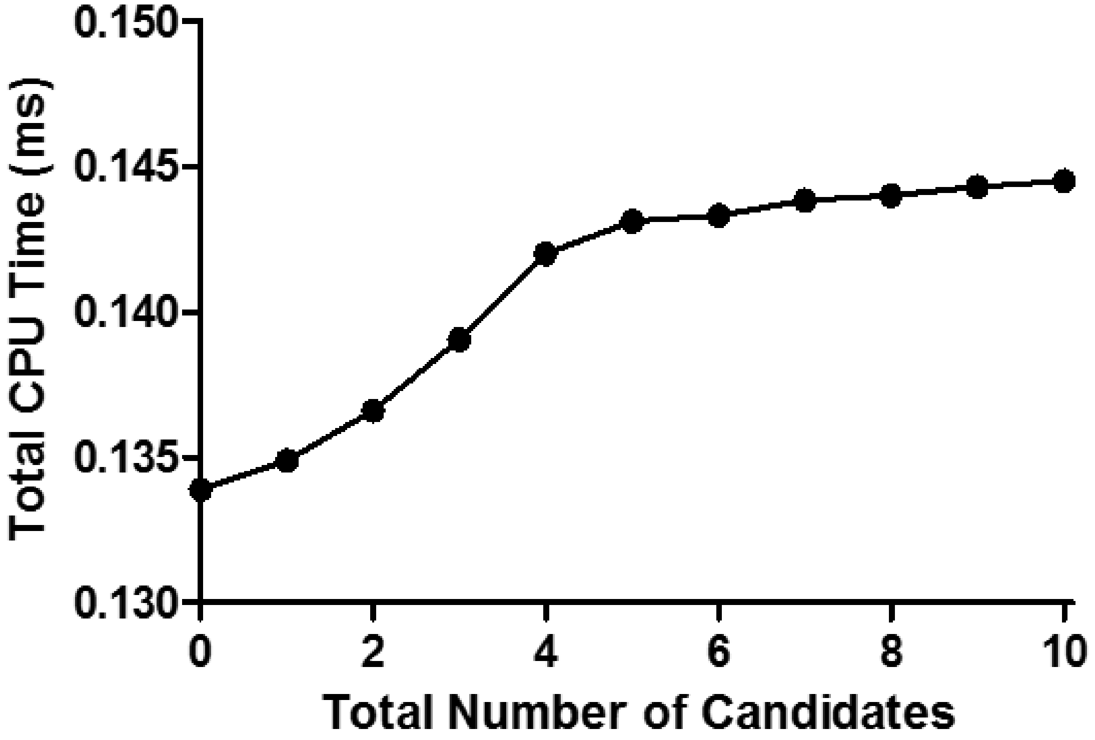


Figure 6.9: No. of candidates vs Processing Time

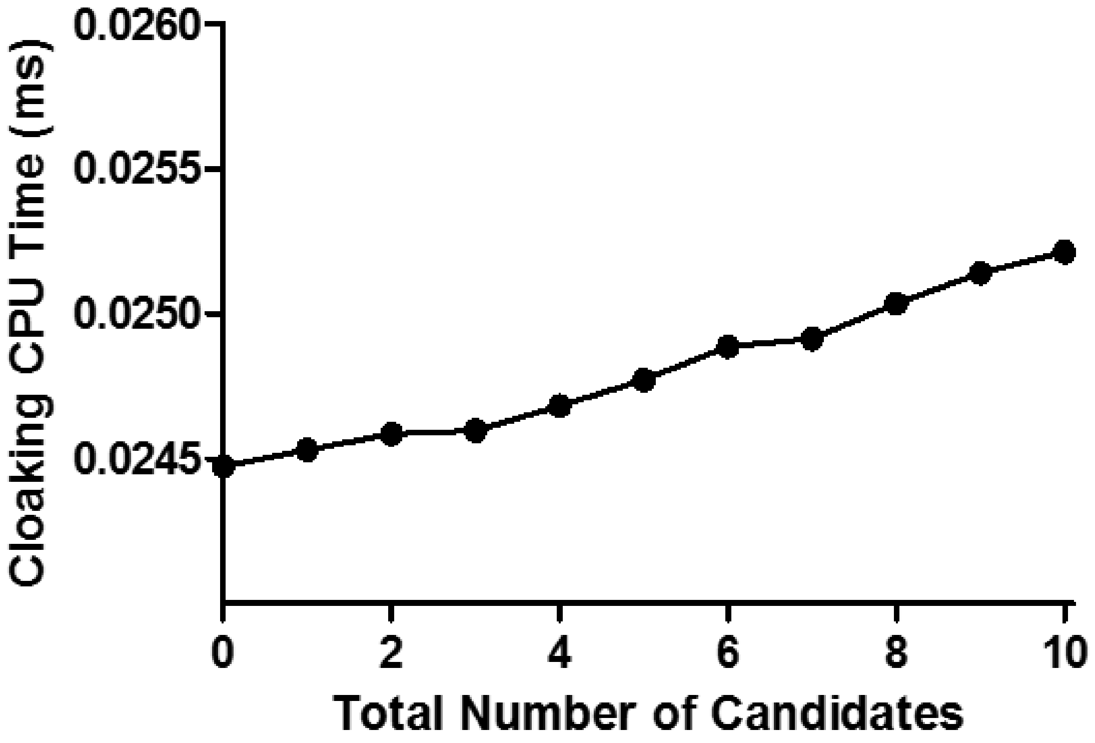


Figure 6.10: No. of candidates vs Processing Time

Figure 6.9 shows the increase in the total CPU time taken by the algorithm from inputting data to returning the list of candidates with the increase in the total number of candidates.

Figure 6.10 shows the increase in the CPU time taken during performing cloaking operations with the increase in the total number of candidates.

Chapter 7

CONCLUSION AND FUTURE WORK

This thesis applied the Casper framework for mobile users to a stationary located object and found it to be equally efficient. The location anonymizer and query processor algorithms were modified to be able to perform efficiently to fit the needs of a single POI whose location does not change. The location anonymizer effectively cloaked the spatial region around the store to fit the store's privacy profile. No dynamic update of the store location was required on a real-time basis. The cloaked region was sent to the query processor, along with a query searching for candidates in the nearby areas, who returned the required list of candidates who might be interested in taking up a job at that store. The experimental results in chapter 5 showed us that the candidate locations returned were accurate. The store may then choose to contact the candidates individually to make progress on hiring new employees. So we are successful in our endeavor to help the store find employees without having to make their location public.

This application of Casper framework to a stationary user base is new and have not been publicly worked on in the past. Many future experimentation can be done related to this application. The candidate locations may be private, instead of public as we considered in our thesis, so experiments can be done on that topic. We can also apply the other types of location privacy algorithms on these algorithms and see how they fare. Extensive experimental analysis of each of these cases can be done to find efficient, accurate and scalable results for this particular scenario.

Bibliography

[1] Mokbel, Mohamed F. and Chow, Chi-Yin and Aref, Walid G. The New Casper: Query Processing for Location Services Without Compromising Privacy. In *VLDB*, 2006.

[2] Brush, A.J. Bernheim and Krumm, John and Scott, James. Exploring End User Preferences for Location Obfuscation, Location-based Services, and the Value of Location. In *ACM*, 2010.

[3] Claudio Agostino Ardagna and Marco Cremonini and Ernesto Damiani and Sabrina De Capitani di Vimercati and Pierangela Samarati. Location Privacy Protection Through Obfuscation-Based Techniques. In DBLP, 2007.

[4] Mohamed F. Mokbel and Chi-Yin Chow and Walid G. Aref. The New Casper: A Privacy-Aware Location-Based Database Server. In *DBLP*, 2007.

[5] Wenyan Zhang and Ximing Cui and Dengfeng Li and Debao Yuan and Mengru Wang. The location privacy protection research in location-based service. In *IEEE*, 2010.

[6] Chi-Yin Chow and Mohamed F. Mokbel and Xuan Liu. Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. In *GeoInformatica*, 2011.

[7] Bao, Jie and Chen, Haiquan and Ku, Wei-Shinn. PROS: A Peer-to-peer System for Location Privacy Protection on Road Networks. In ACM GIS, 2009.

[8] Chow, Chi-Yin and Mokbel, Mohamed F. and Liu, Xuan. A Peer-to-peer Spatial Cloaking Algorithm for Anonymous Location-based Service. In ACM GIS, 2006.

[9] Elmeleegy, Hazem and Ouzzani, Mourad and Elmagarmid, Ahmed and Abusalah, Ahmad. Preserving Privacy and Fairness in Peer-to-peer Data Integration. In ACM SIGMOD, 2010.

[10] Bugra Gedik, Ling Liu. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. In *IEEE*, 2007.

[11] Xun Yi and Russell Paulet and Elisa Bertino and Vijay Varadharajan. Practical Approximate k Nearest Neighbor Queries with Location and Query Privacy. In IEEE, 2016.

[12] Figueroa, Karina and Paredes, Rodrigo. Approximate Direct and Reverse Nearest Neighbor Queries, and the K-nearest Neighbor Graph. IEEE SISAP, 2009.

[13] R. J. Bayardo and R. Agrawal. Data Privacy through Optimal k-Anonymization. In ICDE, 2005.

[14] F. Emekci, D. Agrawal, A. E. Abbadi, and A. Gulbeden. Privacy Preserving Query Processing using Third Parties. In ICDE, 2006.

[15] B. Gedik and L. Liu. A Customizable k-Anonymity Model for Protecting Location Privacy. In ICDCS, 2005.

[16] Minami, Kazuhiro and Borisov, Nikita. Protecting Location Privacy Against Inference Attacks. In ACM, 2010.

[17] Hu, Haibo and Xu, Jianliang and On, Sai Tung and Du, Jing and Ng, Joseph Kee-Yin. Privacy-aware Location Data Publishing. In ACM, 2010.

[18] Zhu, Youwen and Xu, Rui and Takagi, Tsuyoshi. Secure k-NN Computation on Encrypted Cloud Data Without Sharing Key with Query Users. In ACM Cloud Computing, 2013.

[19] Ghinita, Gabriel and Kalnis, Panos and Khoshgozaran, Ali and Shahabi, Cyrus and Tan, Kian-Lee. Private Queries in Location Based Services: Anonymizers Are Not Necessary. In ACM SIGMOD, 2008.

[20] Dave Singelée and Bart Preneel. Location Privacy in Wireless Personal Area Networks. In ACM, 2006.

[21] G. Aggarwal. et al. Vision Paper: Enabling Privacy for the Paranoids. In VLDB, 2004.

[22] R. Agrawal, A. V. Evfimievski, and R. Srikant. Information Sharing Across Private Databases. In SIGMOD, 2003.

[23] W. G. Aref and H. Samet. Efficient Processing of Window Queries in The Pyramid Data Structure. In PODS, 1990.

[24] L. Barkhuus and A. K. Dey. Location-Based Services for Mobile Telephony: a Study of Users Privacy Concerns. In INTERACT, 2003.

[25] M. Duckham and L. Kulik. A Formal Model of Obfuscation and Negotiation for Location Privacy. In Pervasive, 2005.

[26] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In MobiSys, 2003.

[27] U. Hengartner and P. Steenkiste. Protecting Access to People Location Information. In Proceeding of the International Conference on Security in Pervasive Computing, SPC, 2003.

[28] J. I. Hong and J. A. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In MobiSys, 2004.

[29] H. Hu, J. Xu, and D. L. Lee. A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects. In SIGMOD, 2005.

[30] N. Jefferies, C. J. Mitchell, and M. Walker. A Proposed Architecture for Trusted Third Party Services. In the Intl. Conf. on Cryptography: Policy and Algorithms, 1995.

[31] M. F. Mokbel. Towards Privacy-Aware Location-Based Database Servers. In International Workshop on Privacy Data Management, PDM, Apr. 2006.

[32] M. F. Mokbel and W. G. Aref. PLACE: A Scalable Location-aware Database Server for Spatio-temporal Data Streams. IEEE Data Engineering Bulletin, 28(3):310, 2005.

[33] M. F. Mokbel, X. Xiong, and W. G. Aref. SINA: Scalable Incremental Processing of Continuous Queries in Spatio-temporal Databases. In SIGMOD, 2004.

[34] M. F. Mokbel, X. Xiong, W. G. Aref, S. Hambrusch, S. Prabhakar, and M. Hammad. PLACE: A Query Processor for Handling Real-time Spatio-temporal Data Streams (Demo). In VLDB, 2004.

[35] K. Mouratidis, D. Papadias, and M. Hadjieleftheriou. Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring. In SIGMOD, 2005.

[36] M. L. Yiu, N. Mamoulis, and D. Papadias. Aggregate Nearest Neighbor Queries in Road Networks. TKDE, 17(6), 2005.

[37] X. Yu, K. Q. Pu, and N. Koudas. Monitoring K-Nearest Neighbor Queries Over Moving Objects. In ICDE, 2005.